BeatBlendr

Project Summary:

BeatBlendr will be a social media app to connect with people based on their music interests.

Based on the user's **likes** and preferences toward songs we will create a mathematical representation of their music tastes. Using these representations we will recommend other **songs**, **playlists,** and **users** using clustering algorithms and standard distance-based metrics to identify "closeness" between these entities.

Dataset link: https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset

1. **Describe what data is stored in the database. (Where is the data from, and what attributes and information would be stored?)**

The primary source for the data is Kaggle, along with which we plan to create some dummy data for simulating some of our functionalities. We also include a gamification component in the application for crowdsourcing some data and leverage it to increase the accuracy of our system.

These are the primary entities in our database:

Songs: This table will store information about songs like **metadata** which includes song duration, artist, title, genre, and **descriptive** characteristics like acousticness, energy, danceability etc. Some augmented attributes may be added to support some of our proposed features.

Detailed Feature description can be found here:
https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset

Users: This table will store information about users like their Login information (encrypted) and their profile details like name, age, nationality, gender etc. This table will contain simulated dummy records. Some augmented attributes may be added to support some of our proposed features.

Playlists: This table will store information about playlists like the **songs** that they contain and to which **users** these playlists belong. It will include metadata like name, duration, genre etc. Some augmented attributes may be added to support some of our proposed features.

Likes: This table will store information about the **users** liking for **songs**. This table will drive the recommendation components in the application. The attributes will include the referenced user and songs.

Other supporting entities:

User ratings: Rating received by users for a particular song in the gamified component of our app.

User Followers: Musically matched users who are now following each other.

* These are the major relations in our system however, some more relations may be added to enhance and support the proposed.
* Some augmented attributes may be added to the entities to support and enhance some of our proposed features.

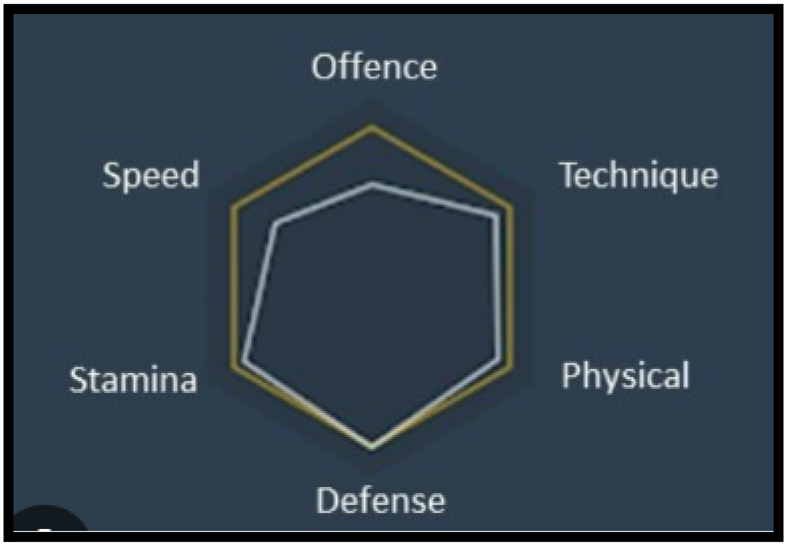
2. **What are the basic functions of your web application? (What can users of this website do? Which simple and complex features are there?)**

Functionalities:

Music recommendation based on mood (Playlist and Song): The user will have to input his mood (eg: happy, party, etc) and based on this we will give him music recommendations. We will establish linkage between various moods and attributes of songs and provide recommendations based on this.

Connecting people based on their playlists: Based on the user profiles (liked songs, favorite genres, favorite artists, etc) we will incorporate a user matching feature. This feature can be used to make friends, meet like minded people, create groups for attending concerts. We will do this using clustering algorithms and standard distance-based metrics to identify “closeness” between these entities.
(Complex)

Song Profile display: We will display the users song profile using a very intuitive and interesting method. Something similar to the image below, with song table attributes at the hexagon vertices.



Music recommendation based on profile: We will recommend songs based on users liked songs and previously played songs. We will assign a score to each song by comparing them with uses liked songs and recommend the top songs.

Mood Analyzer: Analyze users mood based on the songs they are playing. This feature can be extended to add a mood based color scene to the interface. We will use the same mood to song attribute link to develop this feature

Crowd sourcing database by gamification: We plan to implement a functionality which would help in our data set augmentation. We will present the users with an untagged and unclassified song (with no descriptive attributes), the users will listen to the song and provide their ratings for these descriptive attributes in exchange for In Application points which can be redeemed to make In Application purchases like buying a premium subscription. This gamification would benefit the users as they would receive incentives for tagging the songs and it would also improve the BeatBlendr recommendation engine. **(Complex)**

3. **What would be a good creative component (function) that can improve the functionality of your application? (What is something cool that you want to include? How are you planning to achieve it?)**

The 2 creative components that we have envisioned that could make our application stand out will be:

Connecting people based on their playlists:

BeatBlendr will be a social media app to connect with people based on their music interests.

Based on the user's **likes** and preferences toward songs we will create a mathematical representation of their music tastes. Using these representations we will recommend other **songs**, **playlists,** and **users** using clustering algorithms and standard distance-based metrics to identify “closeness” between these entities.

Crowd-sourcing database by gamification:

We plan to augment our **song** dataset by crowdsourcing data with the help of the users of BeatBlendr. New and uncategorized songs will be presented to the users through a mini-game in the application and they will be asked to rate the songs descriptive attributes. Once a certain number of users have rated the song. The song will be added to our dataset. This would not only augment our dataset but also improve the recommendation component and usability of the project.

4. Project Title

BeatBlendr: Your Music Buddy

5. Project Summary:

BeatBlendr is a social networking app for music lovers to share and discover new music and connect with others based on their musical tastes. The application enables users to curate their own playlists based on their likes and encourages them to explore other artists and songs that they have not explored and offer similar musical textures. The application also allows users to connect with fellow music lovers who share the same taste and facilitates collective exploration of the musical landscape.

The application enables this collective exploration and recommendation by using the following features:

Music recommendation based on profile: We will recommend songs based on users liked songs and previously played songs. We will assign a score to each song by comparing them with uses liked songs and recommend the top songs.

Mood Analyzer: Analyze users mood based on the songs they are playing. This feature can be extended to add a mood based color scene to the interface. We will use the same mood to song attribute link to develop this feature

Crowd sourcing database by gamification: We plan to implement a functionality which would help in our data set augmentation. We will present the users with an untagged and unclassified song (with no descriptive attributes), the users will listen to the song and provide their ratings for these descriptive attributes in exchange for In Application points which can be redeemed to make In Application purchases like buying a premium subscription. This gamification would benefit the users as they would receive incentives for tagging the songs and it would also improve the BeatBlendr recommendation engine. **(Complex)**

Music recommendation based on mood (Playlist and Song): The user will have to input his mood (eg: happy, party, etc) and based on this we will give him music recommendations. We will establish linkage between various moods and attributes of songs and provide recommendations based on this.

Connecting people based on their playlists: Based on the user profiles (liked songs, favorite genres, favorite artists, etc) we will incorporate a user matching feature. This feature can be used to make friends, meet like minded people, create groups for attending concerts. We will do this using clustering algorithms and standard distance-based metrics to identify “closeness” between these entities.
(Complex)

Song Profile display: We will display the users song profile using a very intuitive and interesting method. Something similar to the image below, with song table attributes at the hexagon vertices.

6. **Description**

Through this application we are making the process of song discovery easier. We are going to implement a user matching feature using which you can find people who have a similar taste in music. This can help you create your music group and attend live concerts with them

7. **Usefulness**.

In a world filled with an abundance of music streaming services and millions of songs to choose from, finding someone who shares your unique musical taste can be a daunting task. That's where BeatBlendr steps in to change the way people connect, bond, and build relationships through the power of music.

BeatBlendr is an application designed to match users based on their music preferences. Whether you're an avid music enthusiast, a casual listener, or a die-hard fan of a particular genre, BeatBlendr uses recommendation algorithms and user-friendly features to bring like-minded music lovers together.

The benefits of BeatBlendr include having meaningful Connections through music, enhanced music discovery through recommendation features, and shared experiences by collectively curating playlists.

8. **Realness**. Describe what your data is and where you will get it.

We have a comprehensive song dataset which has a total of 21 columns and 114k entries. The important columns include song attributes (multiple columns), artist names, album names, duration of song, popularity, etc. We will create multiple entities using this dataset for implementation of our functionalities.

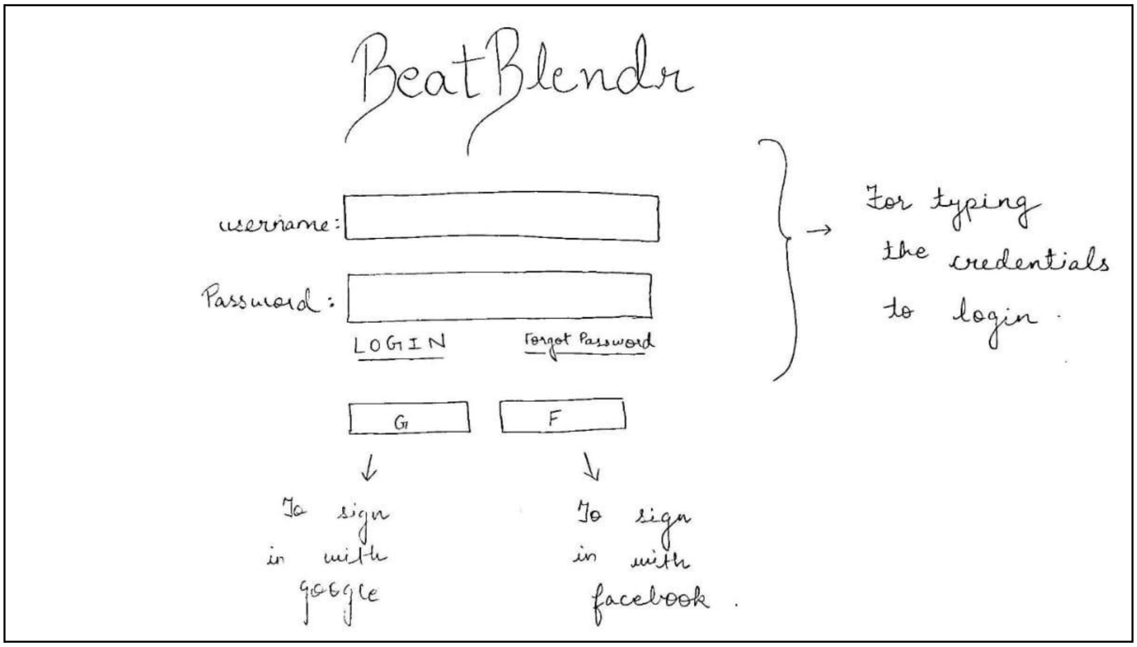
9. **Users interacting with the application (primary operations on the DB)****

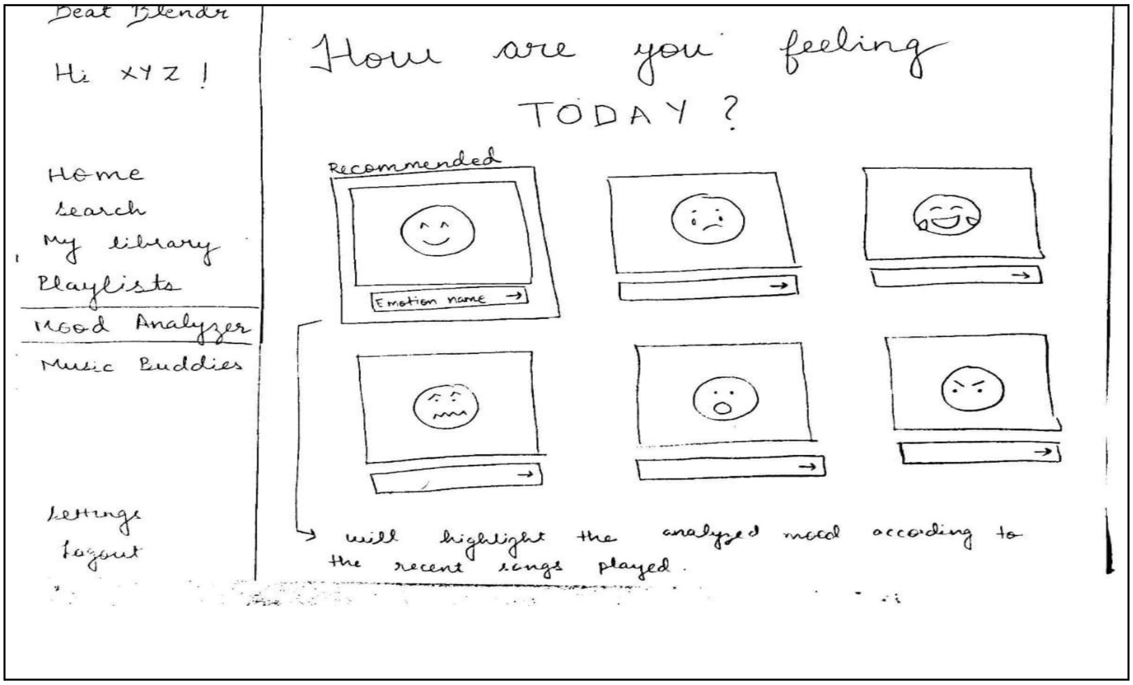
   Users can create their own profiles (INSERT)
   Users can view all the available songs and their liked songs (READ)
   Users can update their ratings for a song (UPDATE)
   Users can create new playlists (SEARCH, INSERT)
   Users can delete their likes for a song (DELETE)
   Users can delete their playlists (DELETE)
   Users can view other users with similar music tastes (READ)

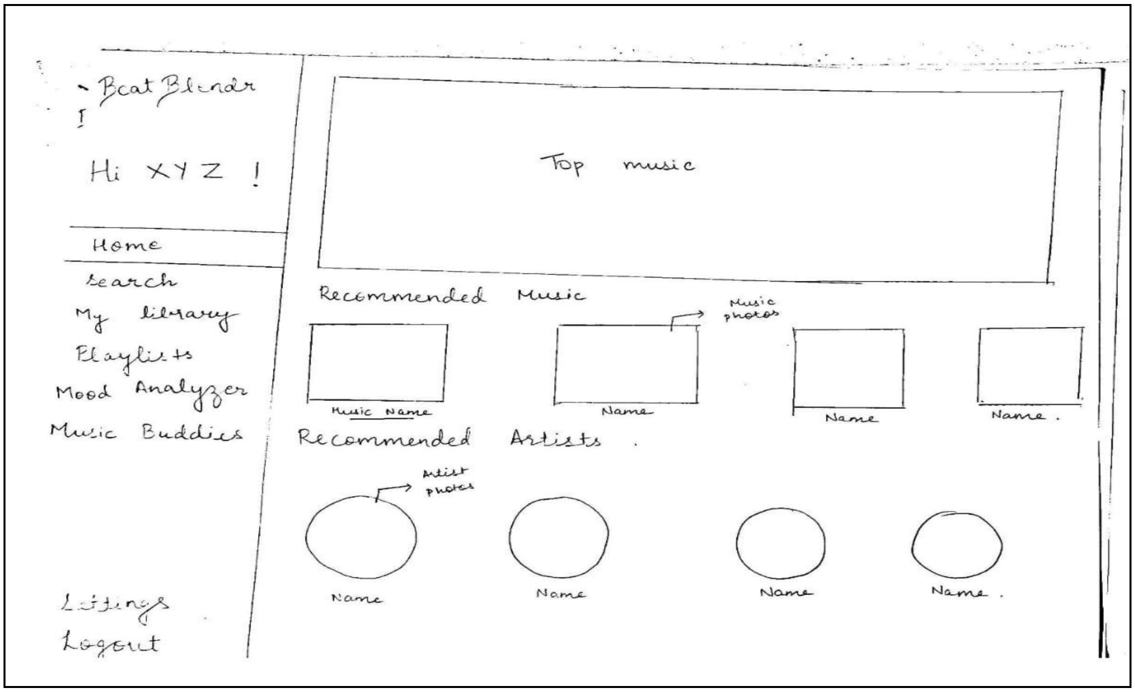
Users can connect with users with similar music tastes (INSERT)

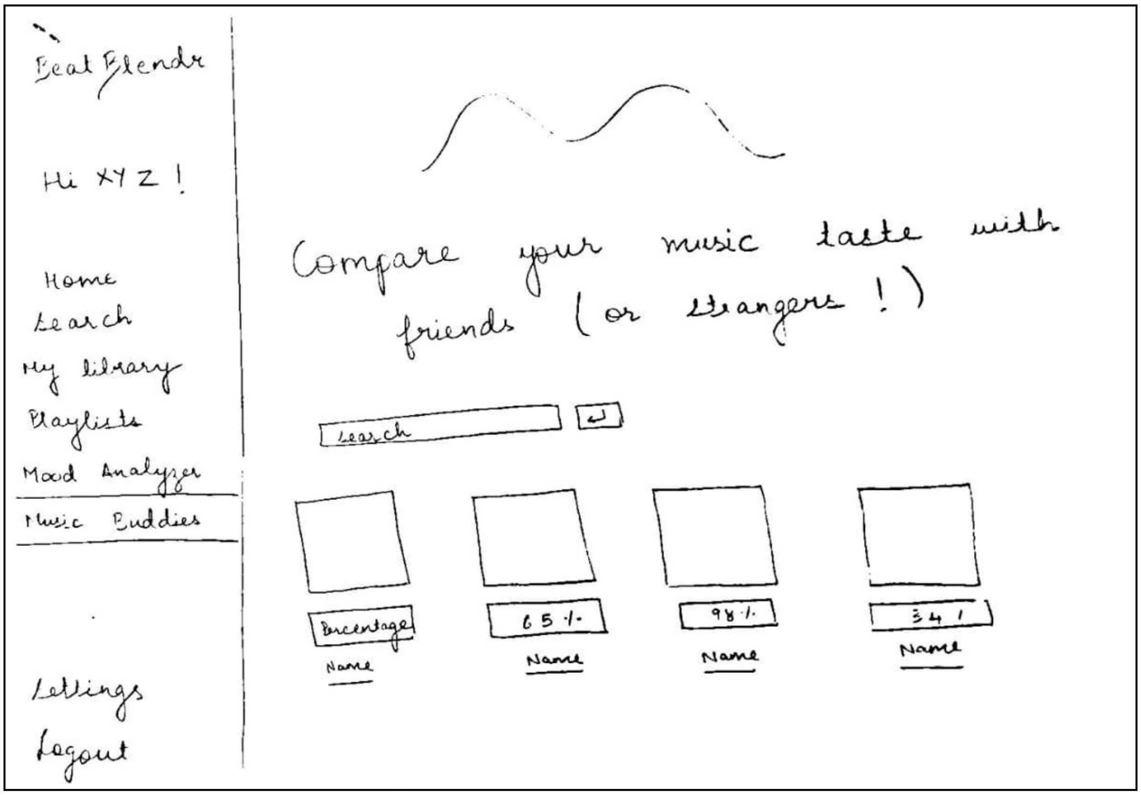
** More DB operations will be added for the recommendation and user-matching features.

9.1 A low-fidelity UI mockup: What do you imagine your final application's interface might look like? A PowerPoint slide or a pencil sketch on a piece of paper works






Beat Blendr
Hi XYZ !
Home
Search
My library
Playlists
Mood Analyzer
Music Buddies
Settings
Logout
Top music
Recommended Music
Music photos
Music Name
Name
Name
Name .
Recommended Artists .
Artist photos
Name
Name
Name
Name .


Beat Blendr
Hi XYZ !
Home
Search
My library
Playlists
Mood Analyzer
Music Buddies
Settings
Logout
Compare your music taste with friends (or strangers !)
Search
↵
Percentage
65%
98%
34 /
Name
Name
Name
Name

9.2 Project work distribution:

Data modeling and normalization:

This part will be done as a group, we will sit together and brainstorm to decide the entities, the relation between them, which tables to create, how to relate the tables and what attributes the tables will contain

Feature development:

We will discuss the features and implementation approach as a group and divide features amongst ourselves. The approximate division is given below

Music recommendation: Shubham Thakar, Shreya Matta

Song Profile display: Shreya Matta, Palvi Shroff

Connecting people based on their playlists: Saumitra Chaskar, Shubham Thakar

Music recommendation based on profile: Palvi Shroff, Saumitra Chaskar

Mood Analyzer: Shubham Thakar, Palvi Shroff

Crowd sourcing database by gamification: Saumitra Chaskar, Shreya Matta